

# Inferring Trust Relationships in Web-based Social Networks

JENNIFER GOLBECK, JAMES HENDLER

University of Maryland, College Park

Computer Science Department

College Park, MD 20742

{golbeck, hendler}@cs.umd.edu

---

The growth of web-based social networking and the properties of those networks have created great potential for producing intelligent software that integrates a user's social network and preferences. Our research looks particularly at assigning trust in web-based social nets and investigates how trust information can be mined and integrated into applications. This paper introduces a definition of trust suitable for use in web-based social networks with a discussion of the properties that will influence its use in computation. We then present two algorithms for inferring trust relationships between individuals that are not directly connected in the network. Both algorithms are shown theoretically and through simulation to produce calculated trust values that are highly accurate.. We then present TrustMail, a prototype email client that uses variations on these algorithms to score email messages in the user's inbox based on the user's participation and ratings in a trust network.

Categories and Subject Descriptors: I.2 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence; H.5 [**INFORMATION INTERFACES AND PRESENTATION**]: Group and Organization Interfaces – *Web-based interaction*

General Terms: Algorithms, Design, Human Factors, Theory

Additional Key Words and Phrases: Social networks, trust, semantic web, online communities, small worlds

---

## 1 INTRODUCTION

In recent years, social networking has moved from the constrained world of academia into the public consciousness. Movies (e.g. "Six Degrees of Separation"), games ("the Kevin Bacon Game" [Oracle of Bacon, 2004], and a new explosion of websites for personal social networking have been released to satisfy the interest in social networks and small worlds. Websites have become particularly popular tools for social experimentation, with millions of members at each of Friendster.com, LinkedIn.com, and Orkut.com to name a few. The Friend-Of-A-Friend (FOAF) project is a semantic web based social network with millions of users whose data is distributed across the web [Friend-of-a-Friend Project, 2004].

| Report Documentation Page  |                                    |                                     | Form Approved<br>OMB No. 0704-0188          |   |                                    |
|--|------------------------------------|-------------------------------------|---|---|------------------------------------|
| Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. |                                    |                                     |   |   |                                    |
| 1. REPORT DATE<br><b>2006</b>  |                                    | 2. REPORT TYPE                      |   | 3. DATES COVERED<br><b>00-00-2006 to 00-00-2006</b> |                                    |
| 4. TITLE AND SUBTITLE<br><b>Inferring Trust Relationships in Web-based Social Networks</b>   |                                    |                                     | 5a. CONTRACT NUMBER                         |   |                                    |
|  |                                    |                                     | 5b. GRANT NUMBER                            |   |                                    |
|  |                                    |                                     | 5c. PROGRAM ELEMENT NUMBER                  |   |                                    |
| 6. AUTHOR(S)   |                                    |                                     | 5d. PROJECT NUMBER                          |   |                                    |
|  |                                    |                                     | 5e. TASK NUMBER                             |   |                                    |
|  |                                    |                                     | 5f. WORK UNIT NUMBER                        |   |                                    |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br><b>University of Maryland, Department of Computer Science, College Park, MD, 20742</b>   |                                    |                                     | 8. PERFORMING ORGANIZATION<br>REPORT NUMBER |   |                                    |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  |                                    |                                     | 10. SPONSOR/MONITOR'S ACRONYM(S)            |   |                                    |
|  |                                    |                                     | 11. SPONSOR/MONITOR'S REPORT<br>NUMBER(S)   |   |                                    |
| 12. DISTRIBUTION/AVAILABILITY STATEMENT<br><b>Approved for public release; distribution unlimited</b>  |                                    |                                     |   |   |                                    |
| 13. SUPPLEMENTARY NOTES<br><b>The original document contains color images.</b>   |                                    |                                     |   |   |                                    |
| 14. ABSTRACT   |                                    |                                     |   |   |                                    |
| 15. SUBJECT TERMS  |                                    |                                     |   |   |                                    |
| 16. SECURITY CLASSIFICATION OF:  |                                    |                                     | 17. LIMITATION OF<br>ABSTRACT               | 18. NUMBER<br>OF PAGES<br><b>41</b>                 | 19a. NAME OF<br>RESPONSIBLE PERSON |
| a. REPORT<br><b>unclassified</b>   | b. ABSTRACT<br><b>unclassified</b> | c. THIS PAGE<br><b>unclassified</b> |   |   |                                    |

The relationships in web-based social networks are more complex than social network models traditionally studied in the social sciences because users can make a variety of assertions about their relationships with others. For example, users may state how well they know the person to whom they are connected or how much they trust that person. These expanded relationships mean that analysis of the networks can take the new information into account to discover more about the nature of the relationships between people. For example, if a network allows users to state the type of relationship they have with another person, this can be used to determine the strength of an indirect connection (e.g. Bob is connected to Frank and the network tells Frank that Bob is the *best friend* of his *roommate* Joe).

This research looks particularly at the concept of trust in social networks. Some networks, like LinkedIn, have trust implied in the network connections. Their motto reflects this: "Find the people you need through the people you trust. " Creating a link to a person on that website implies some amount of business trust for the person. Other networks have a more explicit notion of trust where users assign *trust ratings* to their friends. Orkut allows users to rate one another's trustworthiness with zero to three smiley faces. FOAF has a trust module, developed as part of this project [Golbeck, Hendler, 2004], with thousands of users rating each others' trustworthiness in general or with respect to a given topic on a 1-10 scale.

When trust is explicitly rated on a numerical scale, there should be a way to compose the network data to produce information about the trust between two individuals without a direct connection. The specific problem we look at in this research is whether or not the

data in the network can be used to accurately recommend to one user how much to trust another. With algorithms that can accurately calculate recommendations about trust using data in a social network, it is possible to build socially aware systems where users benefit from their trust and connections.

The goal of our work is to use explicit trust ratings that describe direct connections between people in social networks and compose this information to infer the trust that may exist between two people who are not directly connected. This paper presents two variations on an algorithm to make this calculation in networks where users rate one another on a binary scale (trusted or not trusted). We begin by presenting a definition of trust and illustrating how it fits in with making trust ratings in web-based social networks. For both algorithms, the objective is to infer trust values that are accurate to the person for whom they are calculated. We introduce each algorithm in detail, followed by a theoretical analysis that shows why highly accurate results can be expected. This is reinforced through simulation that demonstrates the correctness in simulated networks. Finally, we demonstrate the potential of using inferred trust values to create trust-aware applications through a prototype of TrustMail, an email client that uses trust ratings as a mechanism to filter email.

## 2 BACKGROUND AND PREVIOUS WORK

Our work is based on the premise that applications will eventually access and utilize the trust data incorporated into web-based social networks. In this section, we introduce the

necessary background for successfully introducing trust to networks on the web, and present some applications that have already begun to take advantage of web-based trust.

## 2.1 Defining Trust

It is an understatement to say that trust is a complicated topic. The word has many subtly different definitions, and it has been the topic of hundreds of books. Putting a computationally usable notion of trust into social networks requires a clear, more narrow definition of the term that still preserves the properties of trust with which we are familiar in our social lives. There are two purposes for which this definition of trust must be clarified: individuals need a definition so they know how to describe their trust for others, and additional features must be understood if trust relationships are to be used in computation that will benefit the user.

In human society, trust depends on a host of factors which cannot be easily modeled in a computational system. Past experience with a person and with their friends, opinions of the actions a person has taken, psychological factors impacted by a lifetime of history and events (most completely unrelated to the person we are deciding to trust or not trust), rumor, influence by others' opinions, and motives to gain something extra by extending trust are just a few of these factors. For trust to be used as a rating between people in social networks, the definition must be focused and simplified.

Marsh [1994] addressed the issue of formalizing trust as a computational concept in his PhD dissertation at the University of Stirling. His model is complex and based on social and psychological factors. Although this work is often cited, the model is highly

theoretical and difficult to implement. It is particularly inappropriate for use in social networks because his focus was on interacting agents that could maintain information about history and observed behaviors. In social networks, users assign a single rating without explicit context or history to their neighbors and thus much of the information necessary for a system like Marsh's is missing.

Deutsch [1962] contains a frequently referenced definition of trust. He states that trusting behavior occurs when a person (say Alice) encounters a situation where she perceives an ambiguous path. The result of following the path can be good or bad, and the occurrence of the good or bad result is contingent on the action of another person (say Bob). Furthermore, the negative impact of the bad result is greater than the positive impact of the good result. This further motivates Alice to make the correct choice. If Alice chooses to go down the path, she has made a trusting choice. She trusts that Bob will take the steps necessary to ensure the good outcome. The requirement that the bad outcome must have greater negative implications than the good outcome has positive implications has been countered in other work [Golombiewski and McConkie, 1975], which does not always require disparity.

Sztompka [1999] presents and justifies a simple, general definition of trust similar to that of Deutsch: "Trust is a bet about the future contingent actions of others." There are two main components of this definition: belief and commitment. First, a person believes that the trusted person will act in a certain way. The belief alone, however, is not enough to say there is trust. Trust occurs when that belief is used as the foundation for making a commitment to a particular action. These two components are also present in the core of

Deutsch's definition: we commit to take the ambiguous path if we believe that the trusted person will take the action that will produce the good outcome.

We adopt this as the definition of trust for our work: *trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a good outcome*. The action and commitment does not have to be significant. We could say Alice trusts Bob regarding email if she chooses to read a message (commits to an action) that Bob sends her (based on her belief that Bob will not waste her time).

## 2.2 Properties of Trust

The primary property of trust that is used in our work is *transitivity*. Trust is not perfectly transitive in the mathematical sense; that is, if Alice highly trusts Bob, and Bob highly trusts Chuck, it does not always and exactly follow that Alice will highly trust Chuck. There is, however, a notion that trust can be passed between people. When we ask a trusted friend for an opinion about a plumber, we are taking the friend's opinion and incorporating that to help form a preliminary opinion of the plumber. In application, the transitivity of trust can work in two ways. First, a person can maintain two types of trust in another person: trust in the person, and trust in the person's recommendations of other people. Alice may trust Bob about movies, but not trust him at all to recommend other people whose opinion about movies is worth considering.

Despite this dichotomy, in social networks it is preferable to let a single value represent both of these ideas. If we say Alice trusts Bob with respect to movies, then her trust in Bob works in two ways. First, it applies to judging movies he recommends. That trust can

also function as trust that extends to judging the trustworthiness of other *people* whom Bob says have trustworthy opinions in movies. Because we expect Bob to make a good recommendation about a film, and Bob trusts Chuck to make good recommendations, Alice can compose those values to develop an idea of Chuck's trustworthiness. This is not to say that Bob's opinion transfers directly to Alice – because she does not know Chuck, she may not trust him as much as Bob does. However, Bob's trust is evidence of shared opinion, and this allows Alice to use Bob's recommendation as evidence that Chuck may also be trustworthy with respect to movies. A single rating system is also more compatible with the traditional way users participate in social networks. Users are rarely, if ever, asked to express opinions of others with such subtle differences.

In social networks it is also important to note the *asymmetry* of trust. For two people involved in a relationship, trust is not necessarily identical in both directions. Because individuals have different experiences, psychological backgrounds, and histories, it is understandable why two people may trust each other different amounts. While asymmetry occurs in all types of human relationships, it is documented more in situations where the two people are not of equal status. For example, employees typically say they trust their supervisors more than the supervisors trust the employees. This is seen in a variety of hierarchies [Yaniv, Kleinberger, 2000]. Even outside of hierarchies, social situations can arise with asymmetric trust. One of the more extreme instances of this is one-way trust, where circumstances force one person to trust the other, but there is no reciprocal trust [Hardin, 2002; Cook, 2001]. Because trust is naturally asymmetric, trust ratings in our system are also asymmetric and represented as directed edges in a network.



One property of trust that is important in social networks, and which has been frequently overlooked in the past, is the *personalization* of trust. Trust is inherently a personal opinion. Two people often have very different opinions about the trustworthiness of the same person. For an example, we need only look to politics. In the United States, when asked "do you trust the current President to effectively lead the country?" the population will be about evenly split – half will trust him very highly, and the other half will have very little trust in his abilities.

Personalization plays into calculating trust recommendations by affecting the accuracy of a recommendation. If a person wants a recommendation about how much to trust the President, an algorithm that simply composes all of the values in the system can be expected to give an answer that falls almost directly in between "very low trust" and "very high trust". With most people having a strong opinion, this middle rating will not mean much. It reflects the opinion of the population, and is not a recommendation to the *individual*. Our algorithm is based on the perspective of the user. It looks at friends who the user trusts about their opinions on a topic, the people whom those friends trust, and so on. Thus, the opinions of people whom the user does not trust much are given very little consideration, and the opinions of people whom the user trusts highly are given more consideration.

Figure 1 depicts a sample social network. The solid lines indicate relationships with trust, and the dashed lines indicate no trust. The node for which we are determining a trust rating, called the *sink*, is trusted by two nodes (8 and 9) and not trusted by two nodes (6 and 7). If a trust rating for the sink were calculated by composing all of the direct ratings of the sink, every node would get the same recommendation. However, if we take into

account the information that we know about the structure of the network from the perspective of each node, a much more informative recommendation can be made. Node 1 can accept information only from it's trusted neighbors. In the end, only the trust ratings given by Nodes 6 and 7 will propagate back to Node 1. Both 6 and 7 do not trust the sink, and only their opinion will be passed back to Node 3 and then to Node 1 who will calculate that the sink is not to be trusted. Similarly, Node 2 also only considers trusted paths. At the end of those paths, Nodes 8 and 9 both have directly rated the sink to be trustworthy. Their values are passed back along the network paths through Nodes 4 and 5 to Node 2. Node 2 will conclude that the sink is to be trusted. Thus, if perspective is taken into account, Node 1 and Node 2 can each receive relevant and accurate information about how much to trust the sink, even though their opinions are diametrically opposed and the information in the network is mixed.

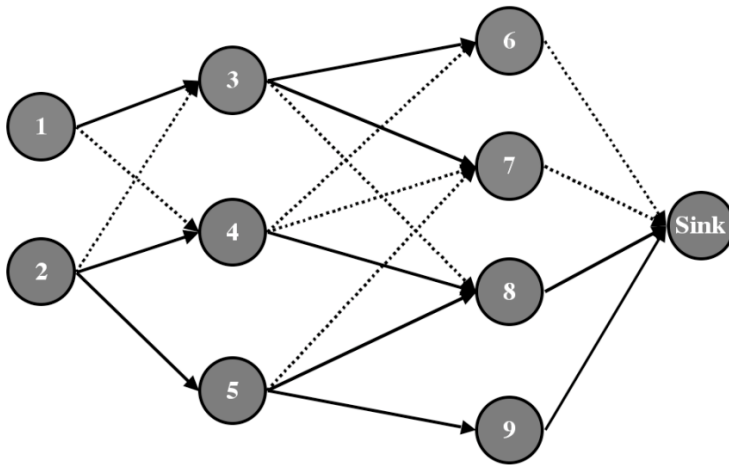


Fig. 1. Nodes consider ratings from the people they trust highly (indicated by solid edges). Nodes with low trust ratings (indicated with dashed edges) are only considered when they are a direct rating of the sink, but are not used in finding paths to the sink. The ratings made by trusted nodes that directly rated the sink are used in coming up with a recommendation about how much the source should trust the sink.

## 2.3 The Values of Trust

The way relationships are expressed in social networks, particularly in web-based social networks, is through an explicit statement. Users rate their connections on a scale usually made available by the service.

There are many systems for rating the how much one person trusts another; in the trust literature, the rating systems vary. The Advogato system uses a three tiered system (Apprentice, Journeyman, Master) for rating its members [Levin, 1998]. Orkut., at <http://Orkut.com>, offers users the ability to rate many features of their friends, including the trustworthiness with zero to three smiley faces. Semantic web trust projects have used a scale of 1-9 [Golbeck, Hendler, 2003], 1-10 or a value that can fall anywhere in the  $[0,1]$  range [Richardson et al., 2003].

For analytic purposes, we first explore networks that only allow a binary rating of either 1 for trustworthy neighbors, or 0 for neighbors who are not trustworthy. Though the  $\{0,1\}$  scale is restrictive, it provides an excellent theoretical foundation for analyzing the effectiveness of the algorithms for inferring trust relationships.

One important note is that a 0 value does not indicate *distrust*. Recall the definition of trust given above: *trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a good outcome*. A 0 value in this system means that the person giving the rating does not believe that the person being rated can be relied upon to take the actions necessary to produce the good outcome. That may be

because the person being rated explicitly tries to do bad things to harm the network, or because of something less sinister like an uncertain user assigning somewhat arbitrary ratings.

## 2.4 Previous Research

The algorithms presented in this paper are designed to be used with any social network, but current implementations are Semantic Web based and use the Friend-Of-A-Friend (FOAF) vocabulary. The FOAF project defines a set of terms for letting users describe people and who they know. FOAF is one of the largest projects on the semantic web, with an estimated 2-5 million users. Some of this data comes from individuals creating their own FOAF files and maintaining that information in their personal web space, but increasingly it is coming from other web-based social networks. LiveJournal (<http://livejournal.com>), eCademy (<http://ecademy.com>) and Tribe (<http://tribe.net>) all publish their users' social network data in FOAF format. Other websites that have gathered social network data have chosen to make those connections available in FOAF format; for example, Howard Dean's Presidential campaign produced thousands of FOAF files representing the social network created when members used a feature of their website to share links with their friends. FOAF has become a recognized means of sharing social network data between social networking websites, and the ease of producing Semantic Web data is encouraging this evolution. The FOAF community is actively rising to this challenge by formalizing their efforts in workshops and online meetings to create a stable core vocabulary that can be used by the widest range of people and applications.

Because it is a Semantic Web ontology, FOAF can be easily extended to capture more detailed personal and relationship information. The Trust Module for FOAF [Golbeck, Hendler 2004] extends the FOAF vocabulary by adding a property where users state how much they trust one another. It has a scale of trust ratings that range from 1 (very little trust) to 10 (very high trust) and is used in several applications [Croucher, 2004; Avestani et al., 2004; Golbeck, Hendler, 2004]. The trust ratings available in this paper are simpler, restricted to a binary scale ("trust" or "no trust"), but these can be modeled using the existing Trust Module for FOAF or with a separate extension of the vocabulary if a binary trust system is valuable to users.

The issue of sharing trust assessments on the semantic web has also been addressed in contexts outside of explicit social networks. Gil and Ratnakar addressed the issue of trusting content and information sources [Gil, 2002] on the Semantic Web. Their TRELLIS system derives assessments about information sources based on individual feedback about the sources. Users of the system can annotate pieces of information and the annotations can include measures of "credibility" and "reliability" about a statement. These are later averaged and presented to the viewer. Using the TRELLIS system, users can view information, annotations (including averages of credibility, reliability, and other ratings), and then make an analysis. Our work uses this notion of augmenting data on the Semantic Web (social network data in our case) with annotations about its trustworthiness.

Once a trust network has been properly modeled and represented, our attention moves to algorithms for calculating recommendations about trust in the network. The question of

trust calculations in social networks has been addressed in several communities with a range of endpoint applications.

The EigenTrust algorithm [Kamvar et al., 2003] is used in peer-to-peer systems and calculates trust with a variation on the PageRank algorithm [Page et al., 1998], used by Google for rating the relevance of web pages to a search. A peer creates a direct trust rating for another peer based on its historical performance. In its simple form, the algorithm uses a matrix representation of the trust values within the system and over a series of iterations it converges to a globally accepted trust rating of each peer. Because of safeguards built into the system, EigenTrust has been shown to be highly resistant to attack. EigenTrust is designed for a peer-to-peer system while ours is designed for use in humans' social networks, and thus there are differences in the approaches to analyzing trust. In the EigenTrust formulation, trust is a measure of performance, and one would not expect a single peer's performance to differ much from one peer to another. Socially, though, two individuals can have dramatically different opinions about the trustworthiness of the same person. Our algorithms intentionally avoid moving toward a global trust value for each individual to preserve the personal aspects that are foundations of social trust.

Raph Levin's Advogato project [Levin, 1998] also calculates a global reputation for individuals in the network, but from the perspective of designated *seeds* (authoritative nodes). His metric composes certifications between members to determine the trust level of a person, and thus their membership within a group. The Advogato website at <http://advogato.org>, for example, certifies users at three levels – apprentice, journeyer, and master. Access to post and edit website information is controlled by these

certifications. Like EigenTrust, the Advogato metric is quite attack resistant. By identifying individual nodes as "bad" and finding any nodes that certify the "bad" nodes, the metric cuts out an unreliable portion of the network. Calculations are based primarily on the good nodes, so the network as a whole remains secure. While the perspective used for making trust calculations is still global in the Advogato algorithm, it is much closer to the methods used in this research. Instead of using a set of global seeds, we let any individual be the starting point for calculations, so each calculated trust rating is given with respect to that person's view of the network.

Richardson et. al.[2003] use social networks with trust to calculate the belief a user may have in a statement. This is done by finding paths (either through enumeration or probabilistic methods) from the source to any node which represents an opinion of the statement in question, concatenating trust values along the paths to come up with the recommended belief in the statement for that path, and aggregating those values to come up with a final trust value for the statement. Current social network systems on the Web, however, primarily focus on trust values between one user to another, and thus their aggregation function is not applicable in these systems. Their paper, intentionally, does not define a specific concatenation function for calculating trust between individuals. The algorithms we define in this paper are aimed specifically at calculating trust between agents, and an exploration of how their algorithms and ours could be combined is an interesting topic for future work.

Ultimately, the goal of this work is to take the trust calculations made by the algorithms that we present in section 4 and incorporate them into applications. Essentially, the inferred trust values are recommendations to one user about how much to trust another

user. There is some research into more traditional recommender systems that suggest our approach to using trust calculations to make recommendations in software systems will be useful. Specifically, there is evidence to support that users will prefer systems with recommendations that rely on social networks and trust relationships over similarity measures commonly used for making recommendations. Research has shown that people prefer recommendations from friends to those made by recommender systems and that users prefer recommendations from systems they trust [Swearingen et al., 2001]. By producing recommendations through the use of trust in social networks, both of those user preferences are addressed. Recommendations come through a network of friends, and are based on the explicit trust expressed by the user.

### 3 GENERATING SOCIAL NETWORKS

Naturally occurring networks take a long time to gain a large number of users, and the topological properties are fixed. To experiment with making trust inferences on networks with various properties, it is necessary to be able to automatically generate network models.

#### 3.1 Building Networks with Correct Topology

It has been widely documented that social networks have the properties of small world networks [Watts, 1999]. The properties of graph structure that define a small world network are connectance and average path length. Connectance (indicated by the variable  $\gamma$ ) is the property of clustering in neighborhoods. Given a node  $n$ , connectance is the fraction of edges between neighbors of  $n$  that actually exist compared to the total number



of possible edges. Small world graphs have strong connectance. The average shortest path length between nodes(indicated with variable  $L$ ) grows logarithmically with the size of the graph in small world networks.

The one difference between the networks in this research and traditional complex systems is that our network has directed edges. Although the  $L$  and  $\gamma$  values are usually calculated with undirected edges, they can easily be calculated with directed edges. The shortest path is calculated by following edges and respecting their direction. Connectance is calculated with twice as many possible edges, since any pair of nodes has two possible directed edges than can connect them.

The work by Watts and Strogatz [Watts, Strogatz, 1998] showed that graphs with small world properties can be generated by randomly rewiring a small number of nodes in a regular graph, like a lattice. The variable  $p$  indicates what percentage of edges should be randomly selected, removed, and randomly reconnected. As  $p$  increases the average path length drops off quickly. The average connectance, on the other hand, remains high until  $p$  gets too large. Creating a lattice and choosing a  $p$  that produces a graph with high connectance and low average path length will produce a small world graph. This model, called the  $\beta$ -model [Watts, 1999] has been shown to successfully emulate the structure of several common social networks, including the co-authorship graph and co-actor graph [Davis et al., 2003; Foster et al., 1963; Newman, 2001; Watts, 1999].

We used a the  $\beta$ -model to create graphs for use in the analysis of the accuracy of algorithms that are presented in section 4. The  $p$  value varied depending on the size and

average degree of our graphs. We verified for each graph size and average degree that the  $p$  value produced graphs with  $L$  and  $\gamma$  values consistent with small world graphs.

We will show two variations of the algorithm. The first uses  $\{0,1\}$  values exclusively – users assign  $\{0,1\}$  ratings, and at every step of the algorithm a node returns  $\{0,1\}$  values. The second algorithm also begins with users assigning  $\{0,1\}$  trust ratings, but as the calculations propagate through the network, nodes return average values that are continuous from  $[0,1]$ . Only in the final step is the value rounded to return either 0 or 1. A careful analysis of these results suggests that similar results can be expected with any scale, but the clarity analysis available with binary ratings allows us to provide insight into *why* other systems may work as well.

### 3.2 Adding Trust Ratings to Graphs

The edges in the generated graphs represent connections between individuals. To generate a trust network, those edges must be augmented with values representing the trust relationship between individuals. This section describes the process of adding trust ratings into a generated graph. The method used for producing these trust values will also form the foundation for analyzing the trust inference algorithms presented in section 4.

One node in the network is randomly chosen as the source. We then give each node a "true" rating. This rating says whether the node is trustworthy (good) or not trustworthy (bad). The number of good and bad nodes are assigned at a pre-determined ratio, and have two properties. First, this true value is treated as *the source's opinion* of the node, as though an all-knowing oracle could tell whether or not the source would consider this

node was good or bad if there were an established relationship. The second property determined by the good/bad rating is the node's behavior. Good nodes agree with the source with a certain probability, while bad nodes always vote incorrectly; the bad nodes will say every good node is bad and every bad node is good.

It is important to note here that we are not studying the *behavior* of the nodes in the network to try to identify "good" or "bad" nodes. Clearly, with bad nodes always voting opposite the source, they would be relatively easy to track down. The bad nodes represent attackers – nodes that may be incorrectly called trustworthy, and can then corrupt the system. While the behavior in these simulations – *always* assigning incorrect values – is worse than we would expect from an attacker in an actual network, it allows us to perform a worst-case analysis of our algorithm since the true value allows us to determine if our inference was correct or incorrect.

Once each node has been given its true value, the trust ratings on the edges are assigned. Bad nodes rate every neighbor opposite its true value. Good nodes rate each neighbor correctly with a certain probability. For example, if we specify that the good nodes are accurate 70% of the time, each neighbor is rated independently with a rating corresponding to the true value with probability 0.7

## 4 MAKING TRUST INFERENCES

In this section we present two algorithms for inferring trust ratings in the generated social networks described in section 3. Since we had the "true" value for each node, we were

able to calculate the recommended trust rating from source to sink, and compare that rating with the "true" value for the sink. This difference is a simple and natural measure of the accuracy of the recommendations made by an algorithm. We show that both theoretically and in simulation the algorithms presented here infer trust ratings that are highly accurate according to the beliefs of the source.

#### 4.1 A Rounding Algorithm

In this algorithm, the source polls each of the neighbors to which it has given a positive reputation rating. Neighbors with zero ("no trust") ratings are ignored, since their reputation means that they give unreliable information. Each of the source's trusted neighbors will return their rating for the sink. The source will then average these ratings and round the final value. This rounded value is the inferred reputation rating from source to sink.

Each of the source's neighbors will use this same process to come up with their reputation ratings for the sink— if there is a direct edge connecting them to the sink, the value of that edge is used; otherwise, the value is inferred. As shown in Figure 2, if a node is encountered in two paths from source to sink, it is considered in each path. Node B and C will both return ratings calculated through D. When a reputation rating from D is first requested, D will average the ratings from E and F. The value is then cached at D, so that the second time D's reputation rating is needed, no calculations are necessary.

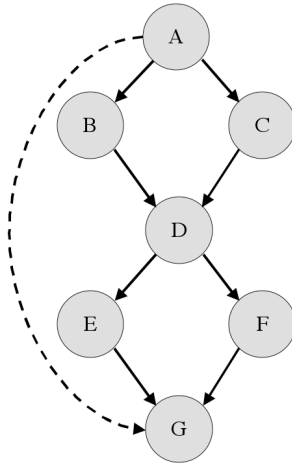


Fig. 2. An illustration of how nodes are used in the inference from node A to node G

## 4.2 Non-Rounding Algorithm

We altered the algorithm presented above by removing the rounding performed by each node before it returns a value. The only rounding is made in one final step, added to the end of the algorithm, where the original source rounds the average of the values returned by its neighbors, so the final inferred value is 0 or 1. Ratings are still assigned as 1 or 0 values (trusted or not trusted). With the algorithmic change, however, intermediate nodes on the path from source to sink return values in the range of  $[0,1]$  instead of returning rounded  $\{0,1\}$  values. Accuracy was determined by taking the difference between the rounded final inferred value and the true value.

## 4.3 Analysis of the Algorithms

There are two variables in the network – percentage of good nodes,  $g$ , and the accuracy,  $p_a$ , of good nodes. The overall accuracy of the direct ratings initially assigned in the network is given by  $g * p_a$ . We call this the *initial accuracy* in the network and represent this with the variable  $a$ . Note that the initial accuracy is a measure of how frequently the

direct ratings of people in the network agree with the true value *according to the source*.

It is not a measure of how accurate a person's ratings are with respect to their own beliefs.

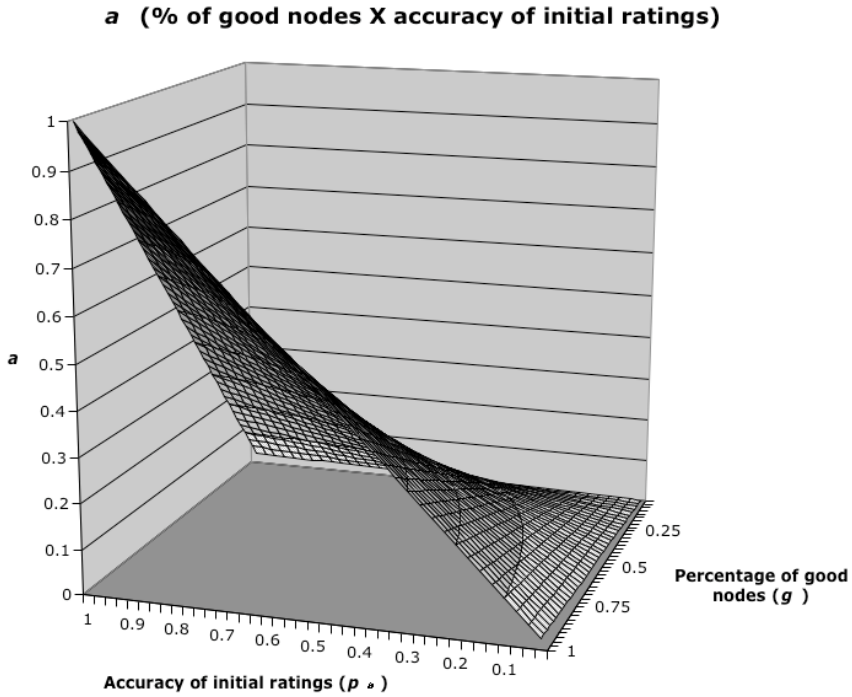


Fig 3. A map of how the initial accuracy in the system changes with  $g$  and  $p_a$ .

By design in these algorithms, a node will make a correct inference if the majority of its neighbors return the correct rating for the sink. Since the bad nodes are always incorrect, the accuracy of the good nodes must compensate to obtain a correct inference from a majority vote. Thus, to obtain a correct inference, the initial accuracy must be at least 0.5.

Let  $a = g * p_a$ . For a given graph with  $n$  nodes, the probability that the majority of the nodes will correctly rate the sink is given by a binomial distribution.

$$\sum_{i=\left\lceil \frac{n}{2} \right\rceil}^n \binom{n}{i} a^i (1-a)^{n-i} \quad (1)$$

The binomial distribution can be approximated by a normal distribution with a mean centered at  $a$ . The Central Limit Theorem says that as  $n$  increases, the binomial distribution gets closer and closer to a normal distribution. That is, the binomial probability of any event approaches the normal probability of the same event. As  $n$  increases, the standard deviation of the normal distribution decreases, making a narrower curve, and thus the probability that a majority of nodes make correct recommendations is closer to the mean  $a$ . Thus, for  $a > 0.5$ , the probability that the mean is greater than 0.5, and ergo the inference is correct approaches 1. Similarly, for  $a < 0.5$ , the probability of a correct inference goes to 0.

$$\lim_{n \rightarrow \infty} \sum_{i=\left\lceil \frac{n}{2} \right\rceil}^n \binom{n}{i} a^i (1-a)^{n-i} \rightarrow 1 \quad \text{for } a > 0.5 \quad (2)$$

Thus, if nodes are accurate at least half of the time, the probability that the recommendation is correct goes to 1. This is a critical point. As long as  $g * p_a$  is greater than half, we can expect to have a highly accurate inference.

This analysis describes one step of rounding. With the non-rounding algorithm, where only the final inferred value is rounded, this analysis applies. In the rounding algorithm, where the average trust value is rounded at each step, the accuracy increases at each step. As the algorithm moves up from the immediate neighbors of the sink toward the source,  $a$  will vary from node to node, but it will increase at each level. Figure 4 illustrates this point where the network starts with all good nodes, accurate in 70% of their classifications. After moving up three levels from the sink, the accuracy of the inference will be approximately 96%.

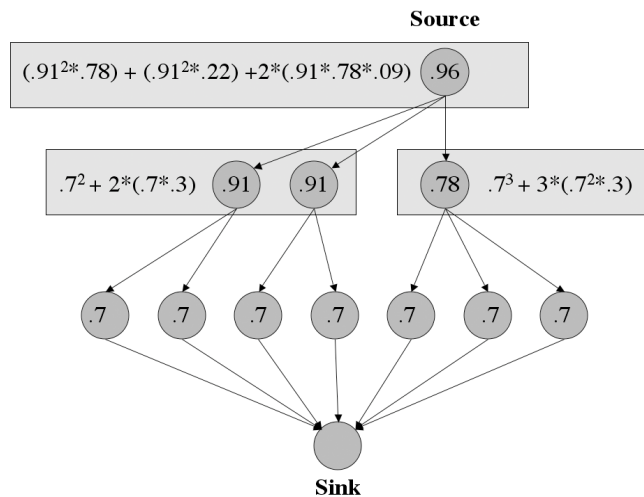


Fig. 4. This figure shows a simple network and demonstrates the increasing probability of accuracy. Beginning with a uniform accuracy of 0.7, the probability of properly classifying the sink increases as we move up the search tree from sink to source. After only three levels, this source has a 96% chance of properly classifying the sink.

This analysis suggests that the rounding algorithm will outperform the non-rounding algorithm. The non-rounding algorithm is an important intermediate step between a system with binary ratings and a system with continuous values. Since continuous values



are used in the intermediate steps between source and sink, this algorithm nearly replicates what would be used when users assign values in a broader range. The analysis here not only shows that the final-step rounding gives good results, but also that accuracy is not lost when the internal rounding is eliminated. Future work will address the shift to a system with continuous values and how a slight variation on the non-rounding algorithm can be effective in such a network.

#### 4.4 Simulations

When inferring the trust rating for a node, the accuracy of the inference is determined by comparing the inferred value to the true value. The simulations presented in this section support the theoretical analysis presented in section 4.3; for both the rounding and non-rounding algorithms, the inferred trust rating is more accurate than the accuracy of the initial trust ratings in the network.

Starting at 0.025 and using increments of 0.025, there are 1,600 pairs  $(g, p_a)$ . For each  $(g, p_a)$  pair we generated 1,000 small-world graphs using the  $\beta$ -model described above. In those graphs, the source and sink were randomly chosen. The trust inference from source to sink made on each graph was checked against the true value of the sink. This experiment was repeated for graphs with 400, 800, and 1600 nodes. Similar results were found for each graph size.

Beginning with the rounding algorithm, experiments showed that the accuracy of the inferred rating was significantly higher than then initial accuracy in the network from the good nodes  $(p_a)$  and the percentage of good nodes  $(g)$ . Figure 5 shows data for the

inferred accuracy using the rounding algorithm on a set of graphs with 400 nodes and an average degree of 16. While the initial accuracy of ratings decreases linearly, the accuracy of the inferred ratings remains higher.

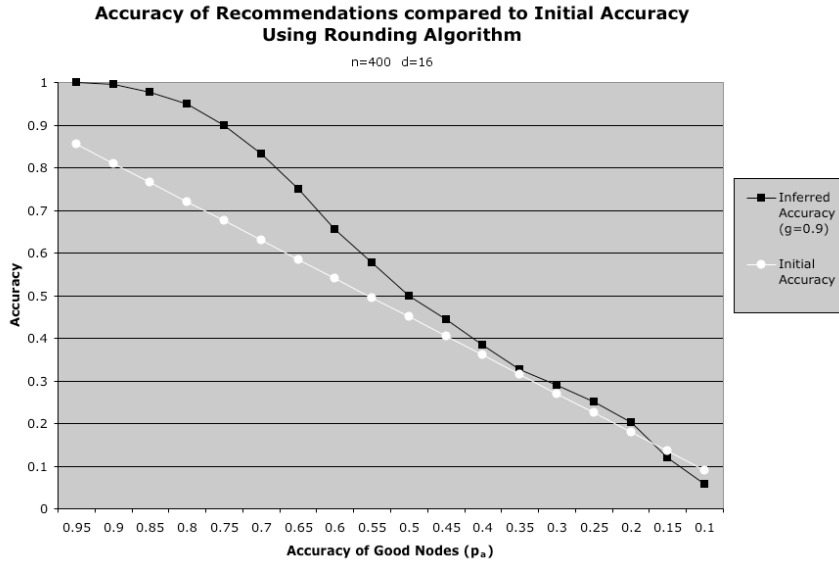


Fig. 5. A comparison of the initial accuracy of trust ratings with the accuracy of inferred ratings using the rounding algorithm for  $n=400$ ,  $d=16$ ,  $g=0.9$ , and a variable  $p_a$ .

In most simulations, the accuracy of a recommendation remains high relative to the initial accuracy until  $p_a * g$  nears 0.5. This is shown in Figure 6.

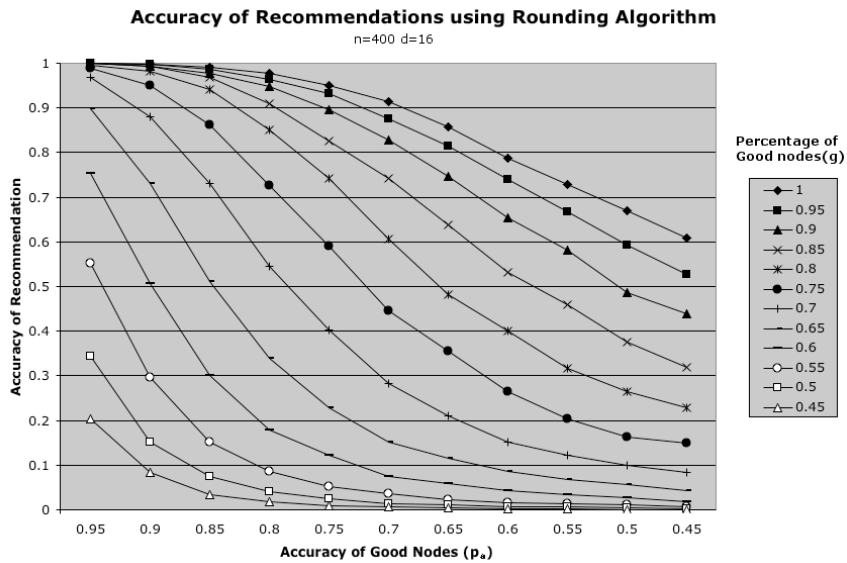


Fig. 6. The accuracy of inferred ratings are shown for various initial percentages of good nodes.

The non-rounding algorithm produced results inline with the theoretical analysis. Even without the intermediate rounding, the inferred values were more accurate than the initial accuracy in the system. The results are less dramatic than for the rounding algorithm, but Figures 7 and 8 shows that the increased accuracy is still present.

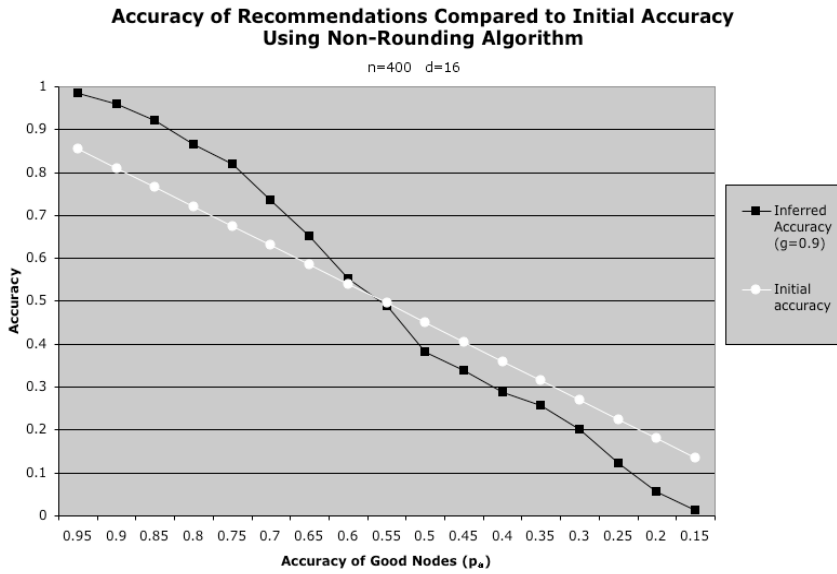


Fig. 7. This figure shows that for  $a > 0.5$ , the inferred accuracy using the non-rounding algorithm is higher than the accuracy of the initial ratings. This is the same effect seen in Figure 5 for the rounding algorithm.

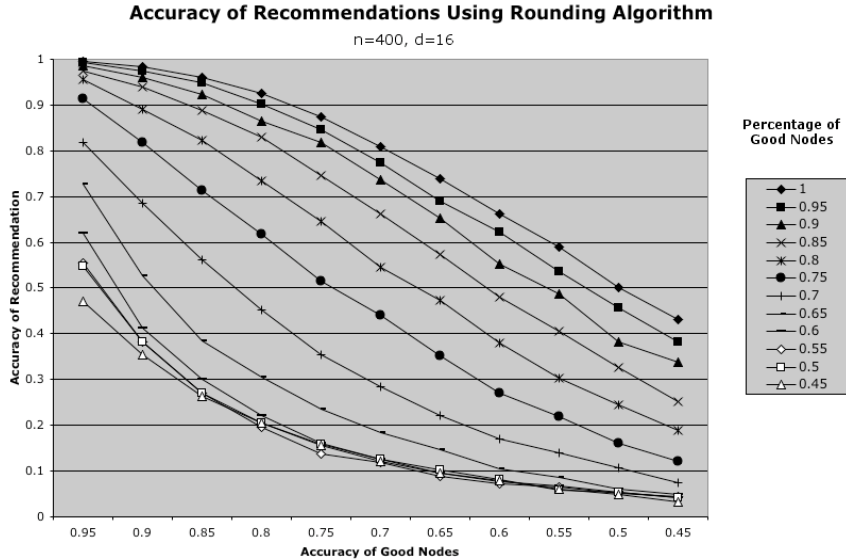


Fig. 8. This figure shows the accuracy of inferred trust values using the non-rounding algorithm for a range of  $g$  and  $p_a$  values. Though less pronounced than the results for the rounding algorithm shown in Figure 6, we can see that the results follow a similar pattern of remaining higher than the  $a$  value for  $a > 0.5$  and lower for  $a < 0.5$ .

To directly compare the two algorithms, Figure 9 shows the accuracy of both the rounding and non-rounding algorithms together for  $g=1$  and  $g=0.9$  with  $p_a > 0.5$ .

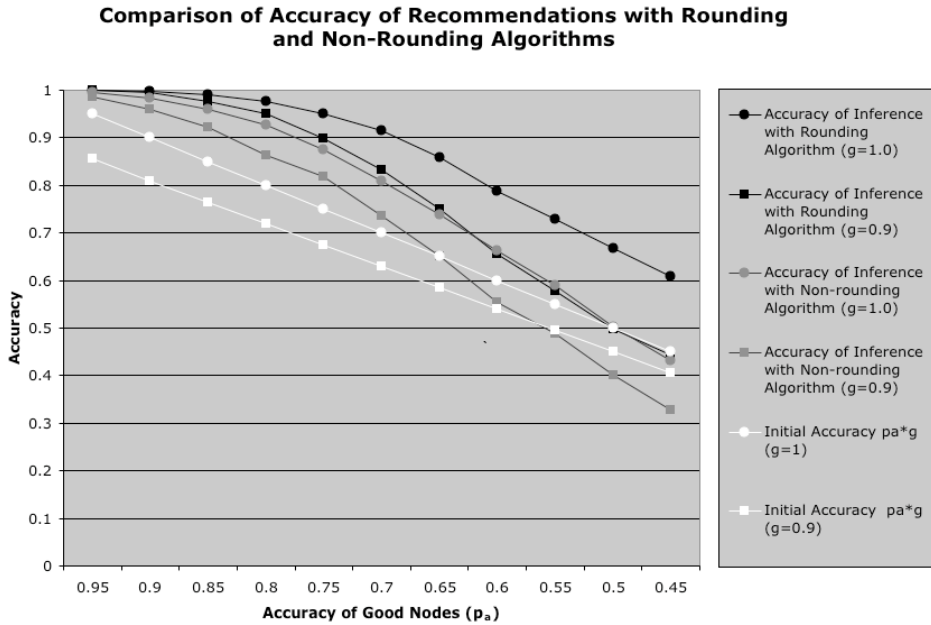


Fig. 9. Comparing the accuracy of trust inferences made with the rounding and non-rounding algorithms.

The results in Figure 9 are drawn from simulations on 400 node networks with average degree of sixteen. The performance follows a similar pattern for both algorithms and both  $g$  values. As the theoretical results would indicate, the rounding algorithm outperforms the non-rounding algorithm for both  $g$  values because the rounding at each step removes more error than is rounded out in the final step of the non-rounding algorithm.

With algorithms that are theoretically and experimentally verified to produce results much more accurate than the initial accuracy in the system, the next step is to integrate those algorithms into applications. The goal of producing accurate algorithms that can calculate trust recommendations from social networks is to use them to create applications that are "socially intelligent". By incorporating users' social preferences into software, features can be tailored to their current connections and preferences. The next section will present one application designed to illustrate the potential of this technique.

## 5 TRUST NETWORKS IN APPLICATIONS: TRUSTMAIL

Consider the case of two research groups working on a project together. The professors that head each group know one another, and each of the professors know the students in their own group. However, neither is familiar with the students from the other group. If, as part of the project, a student sends an email to the other group's professor, how will the professor know that the message is from someone worth paying attention to? Since the name is unfamiliar, the message is not distinguishable from other not-so-important mail in the inbox. This scenario is exactly the type of situation that TrustMail improves upon. The professors need only to rate their own students and the other professor. Since the trust algorithm looks for *paths* in the graph (and not just direct edges), there will be a path from the professor of one research group to the students of the other group through the direct professor to professor link. Thus, even though the student and professor have never met or exchanged correspondence, the student gets a high rating because of the intermediate relationship. If it turns out that one of the students is sending junk type messages, but the network is producing a high rating, the professor can simply add a

direct rating for that sender, downgrading the trust. That will not override anyone else's direct ratings, but will be factored into ratings where the professor is an intermediate step in the path.

## 5.1 Email Filtering with Trust Networks

The fact that spam has become such a ubiquitous problem with email has led to much research and development of algorithms that try to identify spam and prevent it from even reaching the user's mailbox. Many of those techniques have been highly successful, catching and filtering the vast majority of Spam messages that a person receives.

Though work still continues to refine these methods, some focus has shifted to new mechanisms for blocking unwanted messages and highlighting good, or valid, messages. "Whitelist" filters are one of these methods. In these systems, users create a list of approved addresses from which they will accept messages. Any whitelisted messages are delivered to the user's inbox, and all others are filtered into a low-priority folder. These systems do not claim that all of the filtered messages will be spam, but rather that a whitelist makes the inbox more usable by only showing messages from known, valid senders. Though whitelists are nearly 100% effective at blocking unwanted email, there are two major problems cited with them. Firstly, there is an extra burden placed on the user to maintain a whitelist, and secondly, some valid emails will almost certainly be filtered into the low-priority mailbox. If that box contains a lot of spam, the valid messages will be especially difficult to find.

Other approaches have used social networks for message filtering. Boykin and Roychowdhury [2004] create a social network from the messages that a user has

received. Using the structural properties of social networks, particularly the propensity for local clustering, messages are labeled as "spam", "valid", or "unknown" based on clustering thresholds. Their method is able to classify about 50% of a user's email into the spam or valid categories, leaving 50% to be filtered by other techniques.

Our approach takes some of the basic premises of whitelisting and social network-based filtering and extends them. Unlike Boykin and Roychowdhury's technique that builds a social network from the user's own email folders, our technique uses a web-based social network as described in previous sections. Trust ratings – actual or recommended – are used to score messages based on the trust from the recipient to the sender.

The scoring system preserves the whitelist benefit of making the inbox more usable by making "good" messages prominent via high scores. The added benefit is that scores will also appear next to messages from people with whom the user has never had contact before. That is because, if they are connected through a mutual acquaintance in the trust network, we can infer a rating. This diminishes some of the problems with whitelists. There is still a burden for users to create an initial set of trust ratings. However, the properties of the web-based trust network connect users to a much larger population for whom trust values can be calculated. Since scores are available for so many more people than the user explicitly rates, fewer valid messages will be filtered into a low-priority mail folder, lessening the burden on users to find these messages.

It is important to note that the goal of this scoring system is not to give low ratings to bad senders or spam. The main premise is to provide *higher* ratings to *non-spam* senders, so



users are able to identify messages of interest that they might not otherwise have recognized.

Trust scores are not intended to be a stand-alone solution to spam. We envision the mail scoring technique used in conjunction with a variety of other anti-spam mechanisms. There are also some spam issues that particularly effect this algorithm. Forged email headers, where the "From:" line of a message is altered to look like a valid address is one such issue.. Because our technique is designed to identify good messages that make it past spam filters, we do not address the case where a person has a virus sending messages from their account. Our work is not designed to address these problems, and we assume that some other techniques will be designed to deal with them.

## 5.2 The TrustMail Prototype

TrustMail is a prototype email client that adds trust ratings to the folder views of a message. This allows a user to see their trust rating for each individual, and sort messages accordingly. This is, essentially, a message scoring system. The benefit to users is that relevant and potentially important messages can be highlighted, even if the user does not know the sender. The determination of whether or not a message is significant is made using the user's own perspective on the trust network, and thus scores will be personalized to and under the control of each user.

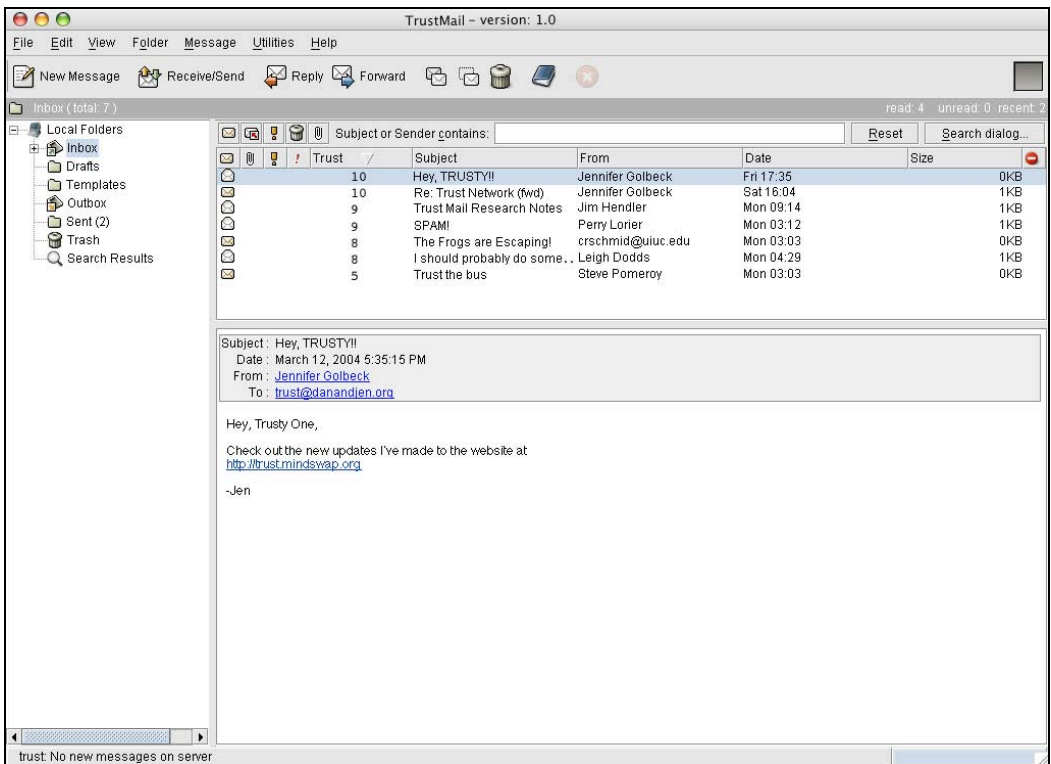


Fig .10. The TrustMail Interface

The algorithms presented earlier in this paper use a binary scale for trust values, but users will want a finer-grained rating system for many applications. In TrustMail, values for rated messages are shown on a user-friendly 1-10 scale, where 1 is a low rating and 10 is a high rating. If the rounding on the final step is removed from the non-rounding algorithm described above, calculated trust values will be returned on a continuous scale and can easily be converted to the 1-10 scale used here (or any other scale that would be useful for an application). Additionally, the same algorithm can be used in a network where users' initial ratings are made on a continuous scale. The continuous initial ratings and continuous inferred ratings give the user a more precise system for understanding trust, but it also does away with the analysis presented in section 4 that demonstrates the

increase in accuracy. Future work will look at how and why algorithms on a continuous scale can be expected to yield accurate results. TrustMail uses this variation on the rating system and the non-rounding algorithm to create a more effective prototype, but the algorithmic assessment is deferred to future work.

The ratings alongside messages are useful, not only for their value, but because they basically replicate the way trust relationships work in social settings. For example, today, it would be sensible and polite for a student emailing a professor she has never met to start the email with some indication of the relationships between the student and the two professors, e.g., “My advisor has collaborated with you on this topic in the past and she suggested I contact you.” The professor may choose to verify the validity of this statement by contacting the student's advisor or finding information that verifies the claim. These ratings are developed by consulting the social network and ratings within it, and serve as evidence of mutual, trusted acquaintances.

TrustMail replaces the process of searching for information about a recipient by utilizing the data in web-based social networks. Because calculations are made from the perspective of the email recipient, high ratings will have necessarily come through people the recipient trusts. This allows our system to complement spam filters by identifying good messages that might otherwise be indistinguishable from unwanted messages, and carrying the validation of a rating drawn from the user's own network of trusted acquaintances.

TrustMail was tested using the trust network developed at <http://trust.mindswap.org>. That Semantic Web-based social network uses the FOAF Trust Module, an ontology that

extends FOAF. Nearly 2,000 members have signed up and entered trust ratings for people they know. The structure of the social network is shown in Figure 11.

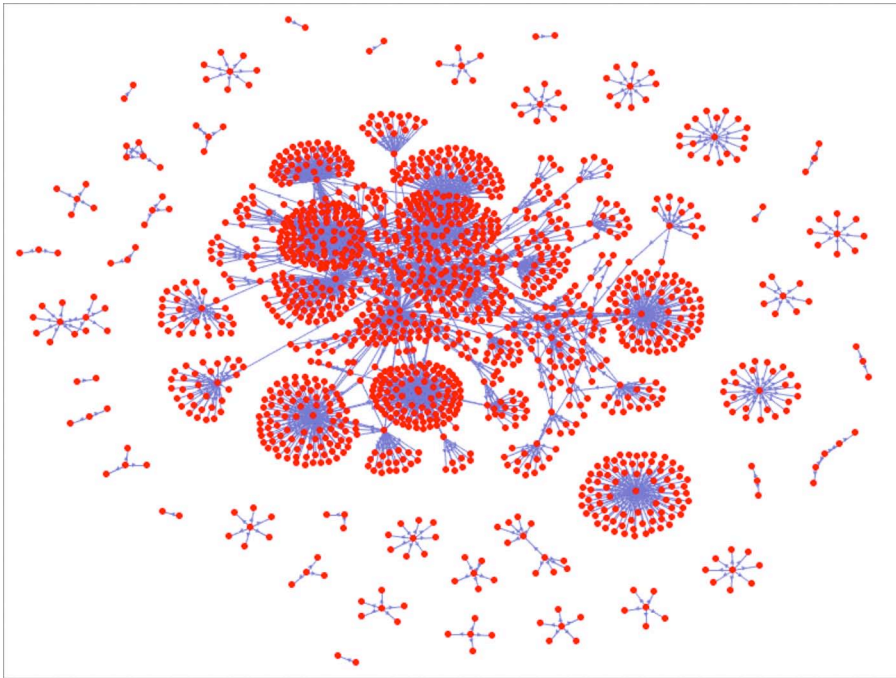


Fig. 11. The trust network at <http://trust.mindswap.org>.

Because the network has a relatively small number of users, we have not thoroughly analyzed on what level TrustMail may affect a user's mailboxes. However, a small preliminary investigation has shown that, for the strongly connected individuals in the trust network, the client is catching and labeling many emails (about 40%) nearly all of which are not spam. Of the emails with trust values, a majority (about 60%) were displayed with inferred ratings as opposed to direct ratings. Future work will expand this study to a large group of users with a more detailed analysis of which emails are

captured, but based on this anecdotal evidence, we expect that users who actively participate in social networks may see significant benefits from these techniques.

## 6 CONCLUSION AND FUTURE WORK

In this paper, we presented the foundations necessary for understanding trust in web-based social networks, and for using trust networks in applications. We presented a definition of trust and illustrated how it could be incorporated into current models of social networks. We then described two algorithms for calculating recommendations about how much one person should trust another based on a personal perspective on the trust network. A theoretical analysis showed that we can expect the recommendations made by these algorithms to be highly accurate relative to the initial accuracy of trust ratings in the system, and this analysis is supported by experimental results. We then described the potential for using these algorithms and their variations in applications to create software that is aware of the user's social preferences by presenting a prototype of TrustMail, an email client that incorporates direct and inferred trust values.

This work used binary trust ratings (trusted or not trusted) for nodes to allow for precise analyses. In reality, trust is not so simple as to be expressed with only two values. A much wider spectrum of values is necessary to capture the complexity of trust. The non-rounding algorithm was a first step in this direction. Nodes assigned values with binary ratings, but values in the range of  $\{0,1\}$  were used as the algorithm passed values back up toward the sink. By changing the initially assigned values from binary to a more continuous system, users will be able to express more nuanced and accurate trust values

for their friends. Future work in this area will require a more theoretical analysis of results with a wider range of values, as well as a comparison with other algorithms.

We are also currently looking at refinements to the algorithms for calculating trust recommendations. These take into account other features of the network's topology that may lead to more accurate values. For example, this work uses a uniform value for  $p_a$  among good nodes. In reality, we can expect that nodes closer to the source will be more likely to agree with it than nodes many steps away. This may mean that limiting the length of paths followed from the source to the sink may lead to better results. In small world networks, the average shortest path between two nodes is logarithmic with the size of the network. This suggests that limiting the search to a short radius around the source will still find a number of paths. Preliminary research in this space is encouraging, and length-limited trust recommendations seem to be more accurate than non-limited ones. We are continuing to work in this area to establish a firmer theoretical and experimental foundation for our early results.

## Acknowledgements

This work, conducted at the Maryland Information and Network Dynamics Laboratory Semantic Web Agents Project, was funded by Fujitsu Laboratories of America -- College Park, Lockheed Martin Advanced Technology Laboratory, NTT Corp., Kevric Corp., SAIC, the National Science Foundation, the National Geospatial-Intelligence Agency, DARPA, US Army Research Laboratory, NIST, and other DoD sources.

Sincere thanks to Bijan Parsia (a highly trusted individual) who influenced, reviewed, and supported this work through every stage of its development. His comments and critiques have been invaluable.

## REFERENCES

Avesani, Paolo, Paolo Massa, Roberto Tiella "Moleskiing: a Trust-aware Decentralized Recommender System" in *Proceedings of 1st Workshop on Friend of a Friend, Social Networking and the (Semantic) Web*. September 1-2, 2004 Galway, Ireland.

Boykin, P. O. & Roychowdhury, V. Personal email networks: an effective anti-spam tool. Preprint, <http://www.arxiv.org/abs/cond-mat/0402143>, (2004).

Cook, Karen (e.d.). 2001. *Trust in Society*, New York: Russell Sage Foundation.

Croucher, Tom, " A model of trust and anonymity in a content rating system for e-learning systems," in *Proceedings of 1st Workshop on Friend of a Friend, Social Networking and the (Semantic) Web*. September 1-2, 2004 Galway, Ireland.

Davis, Gerald, Mina Yoo, Wayne Baker, "The Small World of the American Corporate Elite," *Strategic Organization*, Vol. 1, No. 3, pp. 301-326, August 2003,

Deutsch, Morton. 1962. "Cooperation and Trust. Some Theoretical Notes." in Jones, M.R. (ed) *Nebraska Symposium on Motivation*. Nebraska University Press.

Dudek, C. (2003). "Visual Appeal and the Formation of Trust in E-commerce Web Sites." Unpublished Masters Thesis, Carleton University, Ottawa, Canada.

Foster, C. C., Rapoport, A., and Orwant, C. J. (1963). A study of a large sociogram: Elimination of free parameters. *Behavioural Science* 8:56-65.

The Friend-Of-A-Friend (FOAF) Project (2004). Available: <http://foaf-project.org>.

Gil, Yolanda and Varun Ratnakar, "Trusting Information Sources One Citizen at a Time," *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy, June 2002.

Golbeck, Jennifer, James Hendler, "Reputation Network Analysis for Email Filtering". *Proceedings of the First Conference on Email and Anti-Spam*, July 2004, Mountain View, California.

Golbeck, Jennifer, Bijan Parsia, James Hendler, "Trust Networks on the Semantic Web," *Proceedings of Cooperative Information Agents*, August 27-29, 2003, Helsinki, Finland.

Golbeck, Jennifer, James Hendler, "Accuracy of Metrics for Inferring Trust and Reputation" in *Proceedings of 14th International Conference on Knowledge Engineering and Knowledge Management*, October 5-8, 2004, Northamptonshire, UK.



Golembiewski, Robert T. & McConkie, Mark. 1975. "The Centrality of Interpersonal Trust in Group Processes" in *Theories of Group Processes (Cary Cooper (ed))*. Hoboken, NJ: Wiley.

Hardin, Russell. 2002. *Trust & Trustworthiness*, New York: Russell Sage Foundation.

Kamvar, Sepandar D. Mario T. Schlosser, Hector Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks", *Proceedings of the 12<sup>th</sup> International World Wide Web Conference*, May 20-24, 2003, Budapest, Hungary.

Levin, Raph and Alexander Aiken. "Attack resistant trust metrics for public key certification." *7th USENIX Security Symposium*, San Antonio, Texas, January 1998.

Marsh, S. "Formalising Trust as a Computational Concept." PhD thesis, Department of Mathematics and Computer Science, University of Stirling, 1994.

Newman, M. E. J., "The structure of scientific collaboration networks," *Proc. of the National Academy of Sciences*, Jan 2001; 98: 404 - 409.

The Oracle of Bacon at Virginia (2004): Available <http://oracleofbacon.org>

Orkut (2004). Available: <http://orkut.com>.

Page, L., Brin, S., Motwani, R., & Winograd, T. "The PageRank citation ranking: Bringing order to the web." Technical Report 1998, Stanford University, Stanford, CA.

Richardson, Matthew, Rakesh Agrawal, Pedro Domingos. "Trust Management for the Semantic Web," *Proceedings of the Second International Semantic Web Conference*, 2003. Sanibel Island, Florida.

Swearingen, K. and R. Sinha. "Beyond algorithms: An HCI perspective on recommender systems," *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*, 2001. New Orleans, Louisiana.

Sztompka, Piotr. 1999, *Trust: A Sociological Theory*, Cambridge: Cambridge University Press.

Watts, D. 1999, *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton, NJ: Princeton University Press.

Watts, D. and S. H. Strogatz. "Collective Dynamics of Small-World Networks", *Nature* 393 (1998):440-442.

Yaniv, I., E. Kleinberger, " Advice taking in decision making: Egocentric discounting and reputation formation," *Organizational Behavior and Human Decision Processes*. 2000 Nov; 83(2):260-281